Vulnerability Scan Report

BY

HostedScan Security

GENERATED ON 11/29/2022

Table of Contents

Report Summary

This report contains information on all of the risks found from your vulnerability scans. Each risk is assigned a threat level (high, medium, or low).

Total Risks

Total number of risks found by severity.

2 High	18 Medium	9 Low	0 Accepted

Passive Web Application Vulnerabilities

The OWASP ZAP passive web application scan crawls the pages of a web application. It inspects the web pages as well as the requests and responses sent between the server. The passive scan checks for vulnerabilities such as cross-domain misconfigurations, insecure cookies, vulnerable js dependencies, and more.

Total Risks

Total number of risks found by the passive web application vulnerability scan.

0 High	5 Medium	4 Low	0 Accepted

Risks Summary

Summary of detected risks.

Threat Level	Title	Open Count	Accepted Count
MEDIUM	Absence of Anti-CSRF Tokens	1	0
MEDIUM	Cross-Domain Misconfiguration	1	0
MEDIUM	Vulnerable JS Library	1	0
MEDIUM	Missing Anti-clickjacking Header	1	0
MEDIUM	Referer Exposes Session ID	1	0
LOW	X-Content-Type-Options Header Missing	1	0
LOW	Cookie Without Secure Flag	1	0
LOW	Cookie No HttpOnly Flag	1	0
LOW	Cookie without SameSite Attribute	1	0

Risks Per Target Summary

Breakdown of risk counts for each target.

Target	High	Medium	Low	Accepted
Project https://edistrict.kerala.gov.in	0	5	4	0

Full Risk Details

Detailed information about each risk found by the scan.

Absence of Anti-CSRF Tokens

MEDIUM

Name	Absence of Anti-CSRF Tokens
Description	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Other Information	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "btnSubmit" "fromDate" "hiddenEvent" "rdoID" "rdoName" "toDate" "txtFrom" "txtTo"].

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cross-Domain Misconfiguration

MEDIUM

Name	Cross-Domain Misconfiguration

Description	Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server
Solution	Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance). Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.
Reference	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Other Information	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Vulnerable JS Library

MEDIUM

Name	Vulnerable JS Library
Description	The identified library bootstrap, version 3.3.7 is vulnerable.
Solution	Please upgrade to the latest version of bootstrap.
Reference	https://github.com/twbs/bootstrap/issues/28236 https://github.com/twbs/bootstrap/issues/20184 https://github.com/advisories/GHSA-4p24-vmcr-4gqj
CWE Id	829
WASC Id	-1

Other Information	CVE-2019-8331 CVE-2018-14041 CVE-2018-14040 CVE-2018-14042 CVE-2016-10735

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Missing Anti-clickjacking Header

MEDIUM

Name	Missing Anti-clickjacking Header
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Referer Exposes Session ID

MEDIUM

Name	Referer Exposes Session ID
Description	A hyperlink pointing to another host name was found. As session ID URL rewrite is used, it may be disclosed in referer header to external hosts.

Solution	This is a risk if the session ID is sensitive and the hyperlink refers to an external or third party host. For secure content, put session ID in secured session cookie.
Reference	http://seclists.org/lists/webappsec/2002/Oct-Dec/0111.html
CWE Id	200
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

X-Content-Type-Options Header Missing

LOW

Name	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.
Reference	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Other Information	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cookie Without Secure Flag

LOW

Name	Cookie Without Secure Flag
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cookie No HttpOnly Flag

LOW

Name	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	1004
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes

Project
https://edistrict.kerala.gov.in

Cookie without SameSite Attribute

LOW

Name	Cookie without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Active Web Application Vulnerabilities

The OWASP ZAP active web application scan crawls the pages of a web application. It scans for all of the passive scan checks and additionally makes requests and submits forms to actively test for more vulnerabilities. The active scan checks for vulnerabilities such as SQL injection, remote command execution, XSS, and more.

Total Risks

Total number of risks found by the active web application vulnerability scan.

1 High	6 Medium	4 Low	0 Accepted

Risks Summary

Summary of detected risks.

Threat Level	Title	Open Count	Accepted Count
HIGH	Cross Site Scripting (Reflected)	1	0
MEDIUM	Absence of Anti-CSRF Tokens	1	0
MEDIUM	.htaccess Information Leak	1	0
MEDIUM	Cross-Domain Misconfiguration	1	0
MEDIUM	Vulnerable JS Library	1	0
MEDIUM	Missing Anti-clickjacking Header	1	0
MEDIUM	Session ID in URL Rewrite	1	0
LOW	Cookie No HttpOnly Flag	1	0
LOW	X-Content-Type-Options Header Missing	1	0
LOW	Cookie Without Secure Flag	1	0
LOW	Cookie without SameSite Attribute	1	0

Risks Per Target Summary

Breakdown of risk counts for each target.

Target	High	Medium	Low	Accepted
Project https://edistrict.kerala.gov.in	1	6	4	0

Full Risk Details

Detailed information about each risk found by the scan.

Cross Site Scripting (Reflected)

HIGH

Name	Cross Site Scripting (Reflected)
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.

Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket. Phases: Implementation; Architecture and Design Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies. For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters. Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed. Phase: Architecture and Design For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server. If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated. Phase: Implementation For every web page that is generated, use and specify a character encoding such as ISO-8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-116 for more mitigations related to encoding/escaping. To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set. Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html
CWE Id	79

WASC Id	8
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Absence of Anti-CSRF Tokens

MEDIUM

Name	Absence of Anti-CSRF Tokens
Description	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Other Information	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "btnSubmit" "fromDate" "hiddenEvent" "rdoID" "rdoName" "toDate" "txtFrom" "txtTo"].

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

.htaccess Information Leak

MEDIUM

Name	.htaccess Information Leak

Description	htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.
Solution	Ensure the .htaccess file is not accessible.
Reference	http://www.htaccess-guide.com/
CWE Id	94
WASC Id	14
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cross-Domain Misconfiguration

MEDIUM

Name	Cross-Domain Misconfiguration
Description	Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server
Solution	Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance). Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.
Reference	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Other Information	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Vulnerable JS Library

MEDIUM

Name	Vulnerable JS Library
Description	The identified library bootstrap, version 3.3.7 is vulnerable.
Solution	Please upgrade to the latest version of bootstrap.
Reference	https://github.com/twbs/bootstrap/issues/28236 https://github.com/twbs/bootstrap/issues/20184 https://github.com/advisories/GHSA-4p24-vmcr-4gqj
CWE Id	829
WASC Id	-1
Other Information	CVE-2019-8331 CVE-2018-14041 CVE-2018-14040 CVE-2018-14042 CVE-2016-10735

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Missing Anti-clickjacking Header

MEDIUM

Name	Missing Anti-clickjacking Header
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

CWE Id	1021
WASC Id	15
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Session ID in URL Rewrite

MEDIUM

Name	Session ID in URL Rewrite
Description	URL rewrite is used to track user session ID. The session ID may be disclosed via cross-site referer header. In addition, the session ID might be stored in browser history or server logs.
Solution	For secure content, put session ID in a cookie. To be even more secure consider using a combination of cookie and URL rewrite.
Reference	http://seclists.org/lists/webappsec/2002/Oct-Dec/0111.html
CWE Id	200
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cookie No HttpOnly Flag

LOW

Name	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly

CWE Id	1004
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

X-Content-Type-Options Header Missing

LOW

Name	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.
Reference	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Other Information	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cookie Without Secure Flag

LOW

Name	Cookie Without Secure Flag
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Cookie without SameSite Attribute

LOW

Name	Cookie without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Other Information	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

SSL/TLS Security

The SSLyze security scan checks for misconfigured SSL/TLS certificates, expired certificates, weak ciphers, and SSL/TLS vulnerabilities such as Heartbleed.

Total Risks

Total number of risks found by the SSL/TLS security scan.

0 High	0 Medium	0 Low	0 Accepted

Network Vulnerabilities

The OpenVAS network vulnerability scan tests servers and network connected devices for over 50,000 vulnerabilities.

Total Risks

Total number of risks found by the network vulnerability scan.

1 High	7 Medium	1 Low	0 Accepted

Risks Summary

Summary of detected risks.

Threat Level	Title	CVSS Score	Open Count	Accepted Count
HIGH	Apache Axis2 Document Type Declaration Processing Security Vulnerability	7.5	1	0
MEDIUM	HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8	1	0
MEDIUM	SSL/TLS: Report Weak Cipher Suites	5.0	1	0
MEDIUM	Missing `httpOnly` Cookie Attribute	5.0	1	0
MEDIUM	MacOS X Finder '.DS_Store' Information Disclosure	5.3	1	0
MEDIUM	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	4.3	1	0
MEDIUM	Apache Axis2 <= 1.6.2 Multiple Vulnerabilities	6.4	1	0
MEDIUM	Apache Axis2 engagingglobally Cross-Site Scripting Vulnerability	4.3	1	0
LOW	TCP timestamps	2.6	1	0

Risks Per Target Summary

Breakdown of risk counts for each target.

Target	High	Medium	Low	Accepted
Project https://edistrict.kerala.gov.in	1	7	1	0

Full Risk Details

Detailed information about each risk found by the scan.

Apache Axis2 Document Type Declaration Processing Security Vulnerability

HIGH

nvt

oid	1.3.6.1.4.1.25623.1.0.100814
type	nvt
name	Apache Axis2 Document Type Declaration Processing Security Vulnerability
family	Web application abuses
cvss_base	7.5
severities	**score**: 7.5 **severity**: **type**: cvss_base_v2; **origin**: ; **date**: 2010-09-20T13:31:27Z; **score**: 7.5; **value**: AV:N/AC:L/Au:N/C:P/I:P/A:P
tags	**cvss_base_vector**: AV:N/AC:L/Au:N/C:P/I:P/A:P **summary**: Apache Axis2 is prone to a security vulnerability that may result in information-disclosure or denial-of-service conditions. **insight**: **affected**: The issue affects versions prior to 1.5.2 and 1.6. **impact**: An attacker can exploit this vulnerability to obtain potentially sensitive information by including local and external files on computers running the vulnerable application or by causing denial-of-service conditions. Other attacks are also possible. **solution**: The vendor has released fixes. Please see the references for more information. **vuldetect**: **solution_type**: VendorFix

solution	**–**	The vendor has released fixes. Please see the references for more information.
	type	VendorFix
refs	**ref**	

type	cve
id	CVE-2010-1632

type	url
id	http://www.securityfocus.com/bid/40976

type	url
id	http://ws.apache.org/axis2/

type	url
id	http://geronimo.apache.org/2010/07/21/apache-geronimo-v216-released.html

type	url
id	http://www-01.ibm.com/support/docview.wss?uid=swg27019456

type	url
id	https://issues.apache.org/jira/browse/AXIS2-4450

type	url
id	https://svn.apache.org/repos/asf/axis/axis2/java/core/security/CVE-2010-1632.pdf

type	url
id	http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24027020

type	url
id	http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24027019

type	url
id	http://www.ibm.com/support/docview.wss?uid=swg24027503

type	url
id	http://www.ibm.com/support/docview.wss?uid=swg24027502

type	url
id	http://www-01.ibm.com/support/docview.wss?uid=swg21433581
type	dfn-cert
id	DFN-CERT-2021-0775

threat	High
severity	7.5
qod	value: 80 type:

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

HTTP Debugging Methods (TRACE/TRACK) Enabled

MEDIUM

nvt	oid	1.3.6.1.4.1.25623.1.0.11213
	type	nvt
	name	HTTP Debugging Methods (TRACE/TRACK) Enabled
	family	Web Servers
	cvss_base	5.8
	severities	score: 5.8
		severity: type: cvss_base_v2
		origin:
		date: 2005-11-03T13:08:04Z
		score: 5.8
		value: AV:N/AC:M/Au:N/C:P/I:P/A:N

tags	**cvss_base_vector**	AV:N/AC:M/Au:N/C:P/I:P/A:N
	summary	The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods which are used to debug web server connections.
	insight	It has been shown that web servers supporting this methods are subject to cross-site-scripting attacks, dubbed XST for Cross-Site-Tracing, when used in conjunction with various weaknesses in browsers.
	affected	Web servers with enabled TRACE and/or TRACK methods.
	impact	An attacker may use this flaw to trick your legitimate web users to give him their credentials.
	solution	Disable the TRACE and TRACK methods in your web server configuration. Please see the manual of your web server or the references for more information.
	vuldetect	Checks if HTTP methods such as TRACE and TRACK are enabled and can be used.
	solution_type	Mitigation
solution	**_**	Disable the TRACE and TRACK methods in your web server configuration. Please see the manual of your web server or the references for more information.
	type	Mitigation

refs → **ref**

type	cve
id	CVE-2003-1567
type	cve
id	CVE-2004-2320
type	cve
id	CVE-2004-2763
type	cve
id	CVE-2005-3398
type	cve
id	CVE-2006-4683
type	cve
id	CVE-2007-3008
type	cve
id	CVE-2008-7253
type	cve
id	CVE-2009-2823

type	cve
id	CVE-2010-0386

type	cve
id	CVE-2012-2223

type	cve
id	CVE-2014-7883

type	url
id	http://www.kb.cert.org/vuls/id/288308

type	url
id	http://www.securityfocus.com/bid/11604

type	url
id	http://www.securityfocus.com/bid/15222

type	url
id	http://www.securityfocus.com/bid/19915

type	url
id	http://www.securityfocus.com/bid/24456

type	url
id	http://www.securityfocus.com/bid/33374

type	url
id	http://www.securityfocus.com/bid/36956

type	url
id	http://www.securityfocus.com/bid/36990

type	url
id	http://www.securityfocus.com/bid/37995

type	url
id	http://www.securityfocus.com/bid/9506

type	url
id	http://www.securityfocus.com/bid/9561

type	url
id	http://www.kb.cert.org/vuls/id/867593

	type	url
	id	https://httpd.apache.org/docs/current/en/mod/core.html#traceenable
	type	url
	id	https://techcommunity.microsoft.com/t5/iis-support-blog/http-track-and-trace-verbs/ba-p/784482
	type	url
	id	https://owasp.org/www-community/attacks/Cross_Site_Tracing
	type	cert-bund
	id	CB-K14/0981
	type	dfn-cert
	id	DFN-CERT-2021-1825
	type	dfn-cert
	id	DFN-CERT-2014-1018
	type	dfn-cert
	id	DFN-CERT-2010-0020
threat	Medium	
severity	5.8	
qod	value	99
	type	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

SSL/TLS: Report Weak Cipher Suites

MEDIUM

nvt	oid	1.3.6.1.4.1.25623.1.0.103440
	type	nvt
	name	SSL/TLS: Report Weak Cipher Suites
	family	SSL and TLS
	cvss_base	5.0

severities	**score**	5.0
	severity	**type**: cvss_base_v2 **origin**: **date**: 2012-03-01T16:16:10Z **score**: 5.0 **value**: AV:N/AC:L/Au:N/C:P/I:N/A:N
tags	**cvss_base_vector**	AV:N/AC:L/Au:N/C:P/I:N/A:N
	summary	This routine reports all Weak SSL/TLS cipher suites accepted by a service. NOTE: No severity for SMTP services with 'Opportunistic TLS' and weak cipher suites on port 25/tcp is reported. If too strong cipher suites are configured for this service the alternative would be to fall back to an even more insecure cleartext communication.
	insight	These rules are applied for the evaluation of the cryptographic strength: - RC4 is considered to be weak (CVE-2013-2566, CVE-2015-2808) - Ciphers using 64 bit or less are considered to be vulnerable to brute force methods and therefore considered as weak (CVE-2015-4000) - 1024 bit RSA authentication is considered to be insecure and therefore as weak - Any cipher considered to be secure for only the next 10 years is considered as medium - Any other cipher is considered as strong
	affected	
	impact	
	solution	The configuration of this services should be changed so that it does not accept the listed weak cipher suites anymore. Please see the references for more resources supporting you with this task.
	vuldetect	
	solution_type	Mitigation
solution	**_**	The configuration of this services should be changed so that it does not accept the listed weak cipher suites anymore. Please see the references for more resources supporting you with this task.
	type	Mitigation
refs	**ref**	**type**: cve **id**: CVE-2013-2566 **type**: cve **id**: CVE-2015-2808 **type**: cve **id**: CVE-2015-4000

type	url
id	https://www.bsi.bund.de/SharedDocs/Warnmeldungen/DE/CB/warnmeldung_cb-k16-1465_update_6.html

type	url
id	https://bettercrypto.org/

type	url
id	https://mozilla.github.io/server-side-tls/ssl-config-generator/

type	cert-bund
id	CB-K21/0067

type	cert-bund
id	CB-K19/0812

type	cert-bund
id	CB-K17/1750

type	cert-bund
id	CB-K16/1593

type	cert-bund
id	CB-K16/1552

type	cert-bund
id	CB-K16/1102

type	cert-bund
id	CB-K16/0617

type	cert-bund
id	CB-K16/0599

type	cert-bund
id	CB-K16/0168

type	cert-bund
id	CB-K16/0121

type	cert-bund
id	CB-K16/0090

type	cert-bund
id	CB-K16/0030

type	cert-bund
id	CB-K15/1751

type	cert-bund
id	CB-K15/1591

type	cert-bund
id	CB-K15/1550

type	cert-bund
id	CB-K15/1517

type	cert-bund
id	CB-K15/1514

type	cert-bund
id	CB-K15/1464

type	cert-bund
id	CB-K15/1442

type	cert-bund
id	CB-K15/1334

type	cert-bund
id	CB-K15/1269

type	cert-bund
id	CB-K15/1136

type	cert-bund
id	CB-K15/1090

type	cert-bund
id	CB-K15/1059

type	cert-bund
id	CB-K15/1022

type	cert-bund
id	CB-K15/1015

type	cert-bund
id	CB-K15/0986

type	cert-bund
id	CB-K15/0964

type	cert-bund
id	CB-K15/0962

type	cert-bund
id	CB-K15/0932

type	cert-bund
id	CB-K15/0927

type	cert-bund
id	CB-K15/0926

type	cert-bund
id	CB-K15/0907

type	cert-bund
id	CB-K15/0901

type	cert-bund
id	CB-K15/0896

type	cert-bund
id	CB-K15/0889

type	cert-bund
id	CB-K15/0877

type	cert-bund
id	CB-K15/0850

type	cert-bund
id	CB-K15/0849

type	cert-bund
id	CB-K15/0834

type	cert-bund
id	CB-K15/0827

type	cert-bund
id	CB-K15/0802

type	cert-bund
id	CB-K15/0764

type	cert-bund
id	CB-K15/0733

type	cert-bund
id	CB-K15/0667

type	cert-bund
id	CB-K14/0935

type	cert-bund
id	CB-K13/0942

type	dfn-cert
id	DFN-CERT-2021-0775

type	dfn-cert
id	DFN-CERT-2020-1561

type	dfn-cert
id	DFN-CERT-2020-1276

type	dfn-cert
id	DFN-CERT-2017-1821

type	dfn-cert
id	DFN-CERT-2016-1692

type	dfn-cert
id	DFN-CERT-2016-1648

type	dfn-cert
id	DFN-CERT-2016-1168

type	dfn-cert
id	DFN-CERT-2016-0665

type	dfn-cert
id	DFN-CERT-2016-0642

type	dfn-cert
id	DFN-CERT-2016-0184

type	dfn-cert
id	DFN-CERT-2016-0135

type	dfn-cert
id	DFN-CERT-2016-0101

type	dfn-cert
id	DFN-CERT-2016-0035

type	dfn-cert
id	DFN-CERT-2015-1853

type	dfn-cert
id	DFN-CERT-2015-1679

type	dfn-cert
id	DFN-CERT-2015-1632

type	dfn-cert
id	DFN-CERT-2015-1608

type	dfn-cert
id	DFN-CERT-2015-1542

type	dfn-cert
id	DFN-CERT-2015-1518

type	dfn-cert
id	DFN-CERT-2015-1406

type	dfn-cert
id	DFN-CERT-2015-1341

type	dfn-cert
id	DFN-CERT-2015-1194

type	dfn-cert
id	DFN-CERT-2015-1144

type	dfn-cert
id	DFN-CERT-2015-1113

type	dfn-cert
id	DFN-CERT-2015-1078

type	dfn-cert
id	DFN-CERT-2015-1067

type	dfn-cert
id	DFN-CERT-2015-1038

type	dfn-cert
id	DFN-CERT-2015-1016

type	dfn-cert
id	DFN-CERT-2015-1012

type	dfn-cert
id	DFN-CERT-2015-0980

type	dfn-cert
id	DFN-CERT-2015-0977

type	dfn-cert
id	DFN-CERT-2015-0976

type	dfn-cert
id	DFN-CERT-2015-0960

type	dfn-cert
id	DFN-CERT-2015-0956

type	dfn-cert
id	DFN-CERT-2015-0944

type	dfn-cert
id	DFN-CERT-2015-0937

type	dfn-cert
id	DFN-CERT-2015-0925

type	dfn-cert
id	DFN-CERT-2015-0884

type	dfn-cert
id	DFN-CERT-2015-0881

type	dfn-cert
id	DFN-CERT-2015-0879
type	dfn-cert
id	DFN-CERT-2015-0866
type	dfn-cert
id	DFN-CERT-2015-0844
type	dfn-cert
id	DFN-CERT-2015-0800
type	dfn-cert
id	DFN-CERT-2015-0737
type	dfn-cert
id	DFN-CERT-2015-0696
type	dfn-cert
id	DFN-CERT-2014-0977

threat	Medium
severity	5.0
qod	**value** 98
	type

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Missing `httpOnly` Cookie Attribute

MEDIUM

nvt	**oid**	1.3.6.1.4.1.25623.1.0.105925	
	type	nvt	
	name	Missing `httpOnly` Cookie Attribute	
	family	Web application abuses	
	cvss_base	5.0	
	severities	**score**	5.0
		severity	**type** cvss_base_v2
			origin
			date 2014-09-01T15:00:00Z
			score 5.0
			value AV:N/AC:L/Au:N/C:P/I:N/A:N
	tags	**cvss_base_vector**	AV:N/AC:L/Au:N/C:P/I:N/A:N
		summary	The application is missing the 'httpOnly' cookie attribute
		insight	The flaw is due to a cookie is not using the 'httpOnly' attribute. This allows a cookie to be accessed by JavaScript which could lead to session hijacking attacks.
		affected	Application with session handling in cookies.
		impact	
		solution	Set the 'httpOnly' attribute for any session cookie.
		vuldetect	Check all cookies sent by the application for a missing 'httpOnly' attribute
		solution_type	Mitigation
	solution	**_**	Set the 'httpOnly' attribute for any session cookie.
		type	Mitigation
	refs	**ref**	**type** url
			id https://www.owasp.org/index.php/HttpOnly
			type url
			id https://www.owasp.org/index.php/Testing_for_cookies_attributes_(OTG-SESS-002)
threat	Medium		
severity	5.0		
qod	**value**	80	
	type		

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

MacOS X Finder '.DS_Store' Information Disclosure

MEDIUM

nvt			
	oid	1.3.6.1.4.1.25623.1.0.10756	
	type	nvt	
	name	MacOS X Finder '.DS_Store' Information Disclosure	
	family	Web application abuses	
	cvss_base	5.3	
	severities	**score**	5.3
		severity	**type** cvss_base_v3
			origin NVD
			date 2016-12-20T02:59:00Z
			score 5.3
			value CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
	tags	**cvss_base_vector**	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
		summary	MacOS X creates a hidden file '.DS_Store', in each directory that has been viewed with the 'Finder'. This file contains a list of the contents of the directory, giving an attacker information on the structure and contents of your website.
		insight	
		affected	
		impact	
		solution	Block access to hidden files (starting with a dot) within your webservers configuration
		vuldetect	
		solution_type	Workaround
	solution	**–**	Block access to hidden files (starting with a dot) within your webservers configuration
		type	Workaround

refs	ref	type	cve
		id	CVE-2016-1776
		type	cve
		id	CVE-2018-6470
		type	url
		id	http://www.securityfocus.com/bid/3316
		type	url
		id	http://www.securityfocus.com/bid/3324
		type	url
		id	http://www.securityfocus.com/bid/85054
		type	url
		id	https://helpx.adobe.com/dreamweaver/kb/remove-ds-store-files-mac.html
		type	url
		id	https://support.apple.com/en-us/HT1629
		type	cert-bund
		id	CB-K16/0450
		type	dfn-cert
		id	DFN-CERT-2016-0489

threat	Medium	
severity	5.3	
qod	value	70
	type	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection

MEDIUM

nvt	oid	1.3.6.1.4.1.25623.1.0.117274

type	nvt
name	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection
family	SSL and TLS
cvss_base	4.3

severities

score	4.3

severity

type	cvss_base_v2
origin	
date	2021-03-25T10:41:42Z
score	4.3
value	AV:N/AC:M/Au:N/C:P/I:N/A:N

tags

cvss_base_vector	AV:N/AC:M/Au:N/C:P/I:N/A:N
summary	It was possible to detect the usage of the deprecated TLSv1.0 and/or TLSv1.1 protocol on this system.
insight	The TLSv1.0 and TLSv1.1 protocols contain known cryptographic flaws like: - CVE-2011-3389: Browser Exploit Against SSL/TLS (BEAST) - CVE-2015-0204: Factoring Attack on RSA-EXPORT Keys Padding Oracle On Downgraded Legacy Encryption (FREAK)
affected	All services providing an encrypted communication using the TLSv1.0 and/or TLSv1.1 protocols.
impact	An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection. Furthermore newly uncovered vulnerabilities in this protocols won't receive security updates anymore.
solution	It is recommended to disable the deprecated TLSv1.0 and/or TLSv1.1 protocols in favor of the TLSv1.2+ protocols. Please see the references for more information.
vuldetect	Check the used TLS protocols of the services provided by this system.
solution_type	Mitigation

solution

_	It is recommended to disable the deprecated TLSv1.0 and/or TLSv1.1 protocols in favor of the TLSv1.2+ protocols. Please see the references for more information.
type	Mitigation

refs

ref

type	cve
id	CVE-2011-3389
type	cve
id	CVE-2015-0204
type	url
id	https://ssl-config.mozilla.org/

type	url
id	https://bettercrypto.org/

type	url
id	https://datatracker.ietf.org/doc/rfc8996/

type	url
id	https://vnhacker.blogspot.com/2011/09/beast.html

type	url
id	https://web.archive.org/web/20201108095603/https://censys.io/blog/freak

type	url
id	https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014

type	cert-bund
id	CB-K18/0799

type	cert-bund
id	CB-K16/1289

type	cert-bund
id	CB-K16/1096

type	cert-bund
id	CB-K15/1751

type	cert-bund
id	CB-K15/1266

type	cert-bund
id	CB-K15/0850

type	cert-bund
id	CB-K15/0764

type	cert-bund
id	CB-K15/0720

type	cert-bund
id	CB-K15/0548

type	cert-bund
id	CB-K15/0526

type	cert-bund
id	CB-K15/0509

type	cert-bund
id	CB-K15/0493

type	cert-bund
id	CB-K15/0384

type	cert-bund
id	CB-K15/0365

type	cert-bund
id	CB-K15/0364

type	cert-bund
id	CB-K15/0302

type	cert-bund
id	CB-K15/0192

type	cert-bund
id	CB-K15/0079

type	cert-bund
id	CB-K15/0016

type	cert-bund
id	CB-K14/1342

type	cert-bund
id	CB-K14/0231

type	cert-bund
id	CB-K13/0845

type	cert-bund
id	CB-K13/0796

type	cert-bund
id	CB-K13/0790

type	dfn-cert
id	DFN-CERT-2020-0177

type	dfn-cert
id	DFN-CERT-2020-0111

type	dfn-cert
id	DFN-CERT-2019-0068

type	dfn-cert
id	DFN-CERT-2018-1441

type	dfn-cert
id	DFN-CERT-2018-1408

type	dfn-cert
id	DFN-CERT-2016-1372

type	dfn-cert
id	DFN-CERT-2016-1164

type	dfn-cert
id	DFN-CERT-2016-0388

type	dfn-cert
id	DFN-CERT-2015-1853

type	dfn-cert
id	DFN-CERT-2015-1332

type	dfn-cert
id	DFN-CERT-2015-0884

type	dfn-cert
id	DFN-CERT-2015-0800

type	dfn-cert
id	DFN-CERT-2015-0758

type	dfn-cert
id	DFN-CERT-2015-0567

type	dfn-cert
id	DFN-CERT-2015-0544

type	dfn-cert
id	DFN-CERT-2015-0530

type	dfn-cert
id	DFN-CERT-2015-0396

type	dfn-cert
id	DFN-CERT-2015-0375

type	dfn-cert
id	DFN-CERT-2015-0374

type	dfn-cert
id	DFN-CERT-2015-0305

type	dfn-cert
id	DFN-CERT-2015-0199

type	dfn-cert
id	DFN-CERT-2015-0079

type	dfn-cert
id	DFN-CERT-2015-0021

type	dfn-cert
id	DFN-CERT-2014-1414

type	dfn-cert
id	DFN-CERT-2013-1847

type	dfn-cert
id	DFN-CERT-2013-1792

type	dfn-cert
id	DFN-CERT-2012-1979

type	dfn-cert
id	DFN-CERT-2012-1829

type	dfn-cert
id	DFN-CERT-2012-1530

type	dfn-cert
id	DFN-CERT-2012-1380

type	dfn-cert
id	DFN-CERT-2012-1377

type	dfn-cert
id	DFN-CERT-2012-1292

type	dfn-cert
id	DFN-CERT-2012-1214

type	dfn-cert
id	DFN-CERT-2012-1213

type	dfn-cert
id	DFN-CERT-2012-1180

type	dfn-cert
id	DFN-CERT-2012-1156

type	dfn-cert
id	DFN-CERT-2012-1155

type	dfn-cert
id	DFN-CERT-2012-1039

type	dfn-cert
id	DFN-CERT-2012-0956

type	dfn-cert
id	DFN-CERT-2012-0908

type	dfn-cert
id	DFN-CERT-2012-0868

type	dfn-cert
id	DFN-CERT-2012-0867

type	dfn-cert
id	DFN-CERT-2012-0848

type	dfn-cert
id	DFN-CERT-2012-0838

type	dfn-cert
id	DFN-CERT-2012-0776

type	dfn-cert
id	DFN-CERT-2012-0722

type	dfn-cert
id	DFN-CERT-2012-0638

type	dfn-cert
id	DFN-CERT-2012-0627

type	dfn-cert
id	DFN-CERT-2012-0451

type	dfn-cert
id	DFN-CERT-2012-0418

type	dfn-cert
id	DFN-CERT-2012-0354

type	dfn-cert
id	DFN-CERT-2012-0234

type	dfn-cert
id	DFN-CERT-2012-0221

type	dfn-cert
id	DFN-CERT-2012-0177

type	dfn-cert
id	DFN-CERT-2012-0170

type	dfn-cert
id	DFN-CERT-2012-0146

type	dfn-cert
id	DFN-CERT-2012-0142

type	dfn-cert
id	DFN-CERT-2012-0126

type	dfn-cert
id	DFN-CERT-2012-0123

type	dfn-cert
id	DFN-CERT-2012-0095

type	dfn-cert
id	DFN-CERT-2012-0051

type	dfn-cert
id	DFN-CERT-2012-0047

type	dfn-cert
id	DFN-CERT-2012-0021

type	dfn-cert
id	DFN-CERT-2011-1953

type	dfn-cert
id	DFN-CERT-2011-1946

type	dfn-cert
id	DFN-CERT-2011-1844

type	dfn-cert
id	DFN-CERT-2011-1826

type	dfn-cert
id	DFN-CERT-2011-1774

type	dfn-cert
id	DFN-CERT-2011-1743

type	dfn-cert
id	DFN-CERT-2011-1738

type	dfn-cert
id	DFN-CERT-2011-1706

type	dfn-cert
id	DFN-CERT-2011-1628

type	dfn-cert
id	DFN-CERT-2011-1627

type	dfn-cert
id	DFN-CERT-2011-1619

type	dfn-cert
id	DFN-CERT-2011-1482

threat	Medium
severity	4.3
qod	**value**: 98 **type**

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Apache Axis2 <= 1.6.2 Multiple Vulnerabilities

MEDIUM

nvt	**oid**	1.3.6.1.4.1.25623.1.0.111004
	type	nvt
	name	Apache Axis2 <= 1.6.2 Multiple Vulnerabilities
	family	Web application abuses
	cvss_base	6.4
	severities	**score**: 6.4 **severity**: **type**: cvss_base_v2 **origin** **date**: 2015-03-17T07:00:00Z **score**: 6.4 **value**: AV:N/AC:L/Au:N/C:P/I:P/A:N

tags	**cvss_base_vector**	AV:N/AC:L/Au:N/C:P/I:P/A:N	
	summary	Apache Axis2 is prone to multiple vulnerabilities.	
	insight	The following flaws exist: - a security-bypass vulnerability because the application fails to properly validate SSL certificates from the server. - a security vulnerability involving XML signature wrapping.	
	affected	The issue affects versions up to 1.6.2.	
	impact	Successfully exploiting these issues allows attackers to: - perform man-in-the-middle attacks or impersonate trusted servers, which will aid in further attacks. - may allow unauthenticated attackers to construct specially crafted messages that can be successfully verified and contain arbitrary content. This may aid in further attacks.	
	solution	No known solution was made available for at least one year since the disclosure of this vulnerability. Likely none will be provided anymore. General solution options are to upgrade to a newer release, disable respective features, remove the product or replace the product by another one.	
	vuldetect	Checks if a vulnerable version is present on the target host.	
	solution_type	WillNotFix	
solution	**_**	No known solution was made available for at least one year since the disclosure of this vulnerability. Likely none will be provided anymore. General solution options are to upgrade to a newer release, disable respective features, remove the product or replace the product by another one.	
	type	WillNotFix	
refs	**ref**	**type**	cve
		id	CVE-2012-5785
		type	cve
		id	CVE-2012-4418
		type	cve
		id	CVE-2012-5351
		type	url
		id	http://www.securityfocus.com/bid/56408
		type	url
		id	http://www.securityfocus.com/bid/55508
		type	url
		id	https://issues.apache.org/jira/browse/AXIS2C-1607
threat	Medium		

severity	6.4
qod	**value**: 80 **type**

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Apache Axis2 engagingglobally Cross-Site Scripting Vulnerability

MEDIUM

nvt	**oid**	1.3.6.1.4.1.25623.1.0.111005
	type	nvt
	name	Apache Axis2 engagingglobally Cross-Site Scripting Vulnerability
	family	Web application abuses
	cvss_base	4.3
	severities	**score**: 4.3 **severity**: **type**: cvss_base_v2; **origin**; **date**: 2015-03-17T07:00:00Z; **score**: 4.3; **value**: AV:N/AC:M/Au:N/C:N/I:P/A:N
	tags	**cvss_base_vector**: AV:N/AC:M/Au:N/C:N/I:P/A:N **summary**: Apache Axis2 is prone to a cross-site scripting vulnerability because it fails to properly sanitize user-supplied input. **insight** **affected**: The issue affects versions prior to 1.5.2. **impact**: An attacker may leverage this issue to execute arbitrary script code in the browser of an unsuspecting user in the context of the affected site. This may help the attacker steal cookie-based authentication credentials and to launch other attacks. **solution**: The vendor has released fixes. Please see the references for more information. **vuldetect** **solution_type**: VendorFix

solution	**–**	The vendor has released fixes. Please see the references for more information.
	type	VendorFix
refs	**ref**	**type**	cve
		id	CVE-2010-2103
		type	url
		id	http://www.securityfocus.com/bid/40327
		type	url
		id	http://ws.apache.org/axis2/
		type	url
		id	http://www.exploit-db.com/exploits/12689
threat	Medium		
severity	4.3		
qod	**value**	80	
	type		

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

TCP timestamps

LOW

nvt	**oid**	1.3.6.1.4.1.25623.1.0.80091
	type	nvt
	name	TCP timestamps
	family	General
	cvss_base	2.6

severities	**score**	2.6		
	severity	**type**	cvss_base_v2	
		origin		
		date	2008-10-24T21:33:44Z	
		score	2.6	
		value	AV:N/AC:H/Au:N/C:P/I:N/A:N	
tags	**cvss_base_vector**	AV:N/AC:H/Au:N/C:P/I:N/A:N		
	summary	The remote host implements TCP timestamps and therefore allows to compute the uptime.		
	insight	The remote host implements TCP timestamps, as defined by RFC1323/RFC7323.		
	affected	TCP implementations that implement RFC1323/RFC7323.		
	impact	A side effect of this feature is that the uptime of the remote host can sometimes be computed.		
	solution	To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps		
	vuldetect	Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.		
	solution_type	Mitigation		
solution	**–**	To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime. To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled' Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled. The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment. See the references for more information.		
	type	Mitigation		
refs	**ref**	**type**	url	
		id	http://www.ietf.org/rfc/rfc1323.txt	
		type	url	
		id	http://www.ietf.org/rfc/rfc7323.txt	
		type	url	
		id	https://web.archive.org/web/20151213072445/http://www.microsoft.com/en-us/download/details.aspx?id=9152	
threat	Low			

severity	2.6	
qod	**value**	80
	type	

Vulnerable Target	Accepted	Notes
Project https://edistrict.kerala.gov.in		

Open TCP Ports

The NMAP TCP port scan discovers open ports on with a complete scan of ports 0 to 65535.

Total Risks

Total number of risks found by the TCP port scan.

0 High	0 Medium	0 Low	0 Accepted

Open UDP Ports

The NMAP UDP port scan discovers open ports of common UDP services.

Total Risks

Total number of risks found by the UDP port scan.

0 High	0 Medium	0 Low	0 Accepted